

中華民國經濟部智慧財產局

INTELLECTUAL PROPERTY OFFICE
MINISTRY OF ECONOMIC AFFAIRS
REPUBLIC OF CHINA

茲證明所附文件，係本局存檔中原申請案的副本，正確無訛，

其申請資料如下：

This is to certify that annexed is a true copy from the records of this office of the application as originally filed which is identified hereunder:

申請日：西元 2003 年 04 月 15 日
Application Date

申請案號：092108638
Application No.

申請人：凌陽科技股份有限公司
Applicant(s)

局長
Director General

蔡練生

發文日期：西元 2003 年 7 月 29 日
Issue Date

發文字號：09220766990
Serial No.

※ 申請日期： 92. 4. 15

(英文) _____

肆、中文發明摘要

本發係提出一種可執行條件式指令之處理器及方法，其所執行之指令集包括M位元指令及N位元的指令，該指令集具有條件執行指令及M位元之平行條件執行指令，該平行條件執行指令具有第一N位元指令及第二N位元指令，該處理器包含：一旗標、一指令擷取裝置、一指令解碼裝置、一指令執行裝置及一模式切換裝置，該指令擷取裝置用以擷取所要執行之至少一指令，該指令解碼裝置用以對該擷取之指令進行解碼，該指令執行裝置用以執行該指令解碼裝置所輸出之指令，其中，當執行一條條件執行指令時，依據該條件執行指令之執行結果為條件成立與否，設定該旗標之狀態；以及該模式切換裝置係當該指令擷取裝置所擷取之指令為一平行條件執行指令時，依據該旗標所表示條件成立與否之狀態，切換該指令解碼裝置以對該平行條件執行指令之第一N位元指令或第二N位元指令解碼，以由該指令執行裝置執行之。

伍、英文發明摘要

陸、(一)、本案指定代表圖爲：圖3

(二)、本代表圖之元件代表符號簡單說明：

旗標	310	指令擷取裝置	320
指令解碼裝置	330	指令執行裝置	340
模式切換裝置	350		

柒、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

「無」

捌、聲明事項

☐ 本案係符合專利法第二十條第一項第一款但書或第二款但書規定之期間，其日期為：_____

☐ 本案已向下列國家（地區）申請專利，申請日期及案號資料如下：

【格式請依：申請國家（地區）；申請日期；申請案號 順序註記】

1. 無
2. _____
3. _____

☐ 主張專利法第二十四條第一項優先權：

【格式請依：受理國家（地區）；日期；案號 順序註記】

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____

☐ 主張專利法第二十五條之一第一項優先權：

【格式請依：申請日；申請案號 順序註記】

1. _____
2. _____
3. _____

☐ 主張專利法第二十六條微生物：

☐ 國內微生物 【格式請依：寄存機構；日期；號碼 順序註記】

1. _____
2. _____
3. _____

☐ 國外微生物 【格式請依：寄存國名；機構；日期；號碼 順序註記】

1. _____
2. _____
3. _____

☐ 熟習該項技術者易於獲得，不須寄存。

玖、發明說明

(發明說明應敘明：發明所屬之技術領域、先前技術、內容、實施方式及圖式簡單說明)

【一、發明所屬之技術領域】

本發明係關於處理器的技術領域，尤指一種可執行條件式指令之處理器。

【二、先前技術】

一般處理器在執行一條指令時，會產生條件成立及條件不成立之狀況，依據其結果並利用分枝(branch)或跳躍(jump)指令以執行後續之程式，此種情形因有使用分枝或跳躍指令會使已在管線(pipeline)中的指令被刷新(refresh)以便讀取分枝或跳躍指令目的地之指令，此種方式對具有管線處理之處理器相當無效率。

針對具有管線處理之處理器使用分枝或跳躍指令無效率的問題，於美國第USP5,961,633號專利案公告中，係於指令編碼時使用4位元(第31至第28位元)之條件欄位(condition field)及28位元(第27至第0位元)之操作欄位，一條件測試裝置(condition tester)測試該條件欄位與處理器之4個旗標(N、Z、C、V)，產生一輸出訊號以決定是否放棄該指令，其運作方式如圖1所示，圖1係一C語言程式碼，圖2係圖1中C語言程式經編譯(compile)及組譯(assemble)後之機械碼指令之示意圖，當該處理器執行至指令(1)時，若R1之內含值為0時，該處理器之Z旗標會被設定，當該處理器執行至指令(2)時，指令(2)之條件欄位為EQ，條件測試裝置測試該條件欄位與處理器之Z旗標

相同，故不會產生輸出訊號，所以指令(2)會正常地被該處理器執行，當該處理器執行至指令(6)時，指令(6)之條件欄位為NE，條件測試裝置測試該條件欄位與處理器之Z旗標不相同，故會產生該輸出訊號，所以指令(6)雖會被該處理器執行但其結果會被放棄掉。

處理器執行圖1所示之C語言程式碼時，會執行指令(1)至指令(10)，若R1之內含值為0時，指令(6)至指令(9)之結果會被放棄掉，若R1之內含值不為0時，指令(2)至指令(5)之結果會被放棄掉。

採用此種方法之處理器執行條件指令後，無需依據其結果並用分枝或跳躍指令以執行後續之程式，其可避免因使用分枝或跳躍指令而使已在管線(pipeline)中的指令被刷新(refresh)，可提升具有管線處理之處理器的效率。

然而，採用此種方法之處理器時，其指令編碼時需使用4位元之條件欄位，在16位元指令中只剩12位元可供編碼使用，難以符合一般指令數目之需求，故在16位元指令中無此種條件欄位之設計，同時，無論條件指令之結果為何，後續之指令均需執行，只是有些指令的結果被放棄掉，此亦增加處理器之負擔，因此，習知處理器之條件指令處理方法的設計仍有諸多缺失而有予以改進之必要。

發明人爰因於此，本於積極發明之精神，亟思一種可以解決上述問題之「可執行條件式指令之處理器及其方法」，幾經研究實驗終至完成此項發明。

【三、發明內容】

本發明之目的係在提供一種可執行條件式指令之處理器及其方法，可解決具有管線處理之處理器使用分枝或跳躍指令無效率的問題。並可避免習知技術佔用過多編碼欄位，以及指令不需執行時亦佔用管線處理時間之問題，而達到提高程式碼密度與執行效率的目的。

依據本發明之一特色，係提出一種可執行條件式指令之處理器，其所執行之指令集包括M位元指令及N位元指令（M、N為正整數， $M > N$ ），該指令集具有條件執行指令及M位元之平行條件執行指令，該平行條件執行指令具有第一N位元指令及第二N位元指令，該處理器包含：一旗標、一指令擷取裝置、一指令解碼裝置、一指令執行裝置及一模式切換裝置，該指令擷取裝置用以擷取所要執行之至少一指令；該指令解碼裝置用以對該擷取之指令進行解碼；該指令執行裝置用以執行該指令解碼裝置所輸出之指令，其中，當執行一條件執行指令時，依據該條件執行指令之執行結果為條件成立與否，設定該旗標之狀態；該模式切換裝置係當該指令擷取裝置所擷取之指令為一平行條件執行指令時，依據該旗標所表示條件成立與否之狀態，切換該指令解碼裝置以對該平行條件執行指令之第一N位元指令或第二N位元指令解碼，以由該指令執行裝置執行之。

依據本發明之另一特色，係提出一種於處理器中執行條件式指令之方法，該處理器所執行之指令集包括M位元指令及N位元的指令（M、N為正整數， $M > N$ ），該指令集具有條件執行指令及M位元之平行條件執行指令，該平行條件執行指令具有第一N位元指令及第二N位元指令，該方法主要包含下列步驟：(A)擷取至少一指令，以將其解碼並執行之；(B)當執行一條條件執行指令時，如該條件執行指令之執行結果為條件成立，設定一旗標為作用狀態，如該條件執行指令之執行結果為條件不成立，設定該旗標為清除狀態；以及(C)當擷取之指令為一平行條件執行指令時，如該旗標為作用狀態，對該該平行條件執行指令之第一N位元指令解碼並執行之，如該旗標為清除狀態，則對該平行條件執行指令之第二N位元指令解碼並執行之。

由於本發明設計新穎，能提供產業上利用，且確有增進功效，故依法申請發明專利。

【四、實施方式】

圖3顯示本發明之可執行條件式指令之處理器的方塊圖，其主要包括：一旗標310、一指令擷取裝置320、一指令解碼裝置330、一指令執行裝置340及一模式切換裝置350。該指令擷取裝置320係用以擷取所要執行之至少一指令，其中，此處理器所執行之指令集包括M位元指令及N位元的指令（M、N為正整數， $M > N$ ，例如 $M = 32$ 且 $N = 16$ ），於處理器之指令集中，除了一般性之M位元

及N位元指令外，尚包括有N位元或M位元之條件執行指令（例如比較指令）、及M位元之平行條件執行指令，而平行條件執行指令係為具有至少兩N位元指令之M位元指令，如圖4所示，一32位元之平行條件執行指令包括第一N（ $N=16$ ）位元指令及第二N（ $N=16$ ）位元指令，其可依據條件執行指令之執行結果決定執行第一N位元指令或第二N位元指令。

該指令解碼裝置330係用以對擷取之指令進行解碼，指令執行裝置340則用以執行該指令解碼裝置所解碼輸出之指令，而如果所執行之指令為一N位元或M位元條件執行指令時，該指令執行裝置340將依據該條件執行指令之執行結果，設定該旗標310之狀態，亦即，當該條件執行指令之執行結果為條件成立時，將旗標310之狀態設為”真”，而當該條件執行指令之執行結果為條件不成立時，將旗標310之狀態設為”假”。

該模式切換裝置350用以切換處理器在執行一平行條件執行指令時之模式，其中，當該指令擷取裝置320所擷取之指令為一平行條件執行指令時，該模式切換裝置350依據該旗標310所表示條件成立與否之狀態，來切換該指令解碼裝置330以對該平行條件執行指令之第一N位元指令或第二N位元指令解碼，亦即，當旗標310之狀態為”真”時，該指令解碼裝置330係對該平行條件執行指令之第一N位元指令解碼，而該指令執行裝置340則執行此一第一N位元指令，而當旗標310之狀態為”假”時，該指令解碼裝置330係對該平行條件執行指令之第二N位元指

令解碼，而該指令執行裝置340則執行此一第二N位元指令。

圖5顯示本發明之一實際範例，其係將圖1之C語言程式碼經編譯(compile)及組譯(assemble)後之機械碼指令之示意圖，其中，指令(1)為一M位元($M=32$)之條件執行指令(比較指令)，而當該處理器執行至指令(1)時，若暫存器R1之內含值為0時，比較之結果相同，因此，條件執行指令之執行結果為條件成立，故該旗標310會被設定為真，當該處理器執行至平行條件執行指令(2)時，該處理器判別該旗標為真，故僅執行第一N位元指令[MOVEQ R1, R5]，而不執行第二N位元指令[MOVNE R1, R9]，同理，對於隨後的平行條件執行指令(3)~(5)，因旗標310已被設定為真，該處理器僅執行第一N位元指令[MOVEQ R2, R6]、[MOVEQ R3, R7]、[MOVEQ R4, R8]指令，之後，已無平行條件執行指令，該處理器繼續執行一般之M位元指令(6)。

而若該處理器執行至指令(1)時，暫存器R1之內含值不為0時，比較之結果不相同，因此，條件執行指令之執行結果為條件不成立，故該旗標310會被設定為假，因此，該處理器執行至平行條件執行指令(2)~(5)時，該處理器判別該旗標為假，故僅執行第二N位元指令[MOVEQ R1, R9]、[MOVEQ R1, R10]、[MOVEQ R1, R11]、[MOVEQ R1, R12]，之後，已無平行條件執行指令，該處理器繼續執行一般之M位元指令(6)。

圖 6 係本發明之另一範例之示意圖，其中，在條件執行指令(指令(1))與平行條件執行指令(指令(3))之間可以存有不影響旗標之其他指令，該處理器於執行指令(1)時，依其執行結果設定該旗標 310，由於指令(2)並不影響該旗標，故該處理器仍可依據該旗標 310 而選擇執行平行條件執行指令(3)至(6)中的第一 N 位元指令或第二 N 位元指令。

圖 7 係本發明之再一範例之示意圖，其中，在平行條件執行指令之間可以存有不影響旗標之其他指令，該處理器於執行指令(1)時，依其執行結果設定該旗標，由於指令(4)並不影響該旗標，故該處理器仍可依據該旗標而選擇執行平行條件執行指令(5)至(6)中之第一 N 位元指令(MOVEQ R3, R7)或第二 N 位元指令(MOVNE R1, R11)。

圖 8 係本發明之又一範例之示意圖，其顯示該條件執行指令為一 N 位元 ($N=16$) 指令，處理器於執行指令(1)中條件執行指令(CMP R1, 0)時，依其執行結果設定該旗標，由於指令(1)中的其他指令(other instruction)並不影響該旗標，故該處理器仍可依據該旗標而選擇執行平行條件執行指令(2)至(5)中之第一 N 位元指令(MOVEQ R1, R5)或第二 N 位元指令(MOVNE R1, R9)。

由上述之說明可知，本發明之技術無需像習知技術使用 4 位元條件欄位(condition field)，而不會浪費許多指令編碼空間，也可用較短指令碼來編碼條件指令之後續執行指令，而提高程式編碼密度(Code Density)。本發明

在執行如圖1之程式時，僅需花費6個時序(clock)，遠較習知技術需花費10個時序為少，其無需浪費多餘的指令週期在放棄執行結果的指令上，故其執行效能遠較習知技術更好。

綜上所陳，本發明無論就目的、手段及功效，在在均顯示其迥異於習知技術之特徵，實為一極具實用價值之發明，懇請貴審查委員明察，早日賜准專利，俾嘉惠社會，實感德便。惟應注意的是，上述諸多實施例僅係為了便於說明而舉例而已，本發明所主張之權利範圍自應以申請專利範圍所述為準，而非僅限於上述實施例。

【五、圖式簡單說明】

圖1：係一C語言程式。

圖2：係圖1中C語言程式經習知技術編譯及組譯後之機械碼指令之示意圖。

圖3：係本發明之一種可執行條件式指令之處理器之架構圖。

圖4：係本發明之平行條件執行指令的格式。

圖5：係圖1中C語言程式經本發明技術編譯及組譯後之機械碼指令之示意圖。

圖6：係本發明之另一實例之示意圖。

圖7：係本發明之再一實例之示意圖。

圖8：係本發明之又一實例之示意圖。

【圖號說明】

旗標	310	指令擷取裝置	320
指令解碼裝置	330	指令執行裝置	340
模式切換裝置	350		

拾、申請專利範圍

1. 一種可執行條件式指令之處理器，其所執行之指令集包括M位元指令及N位元的指令（M、N為正整數， $M > N$ ），該指令集具有條件執行指令及M位元之平行條件執行指令，該平行條件執行指令具有第一N位元指令及第二N位元指令，該處理器包含：

一旗標；

一指令擷取裝置，用以擷取所要執行之至少一指令；

一指令解碼裝置，用以對該擷取之指令進行解碼；

一指令執行裝置，用以執行該指令解碼裝置所輸出之指令，其中，當執行一條條件執行指令時，依據該條件執行指令之執行結果為條件成立與否，設定該旗標之狀態；以及

一模式切換裝置，其當該指令擷取裝置所擷取之指令為一平行條件執行指令時，依據該旗標所表示條件成立與否之狀態，切換該指令解碼裝置以對該平行條件執行指令之第一N位元指令或第二N位元指令解碼，以由該指令執行裝置執行之。

2. 如申請專利範圍第1項所述之處理器，其中，當該指令執行裝置執行一條條件執行指令、且條件成立時，設定該旗標為第一邏輯狀態，而當該指令執行裝置執行一條條件執行指令、且條件不成立時，設定該旗標為第二邏輯狀態。

3. 如申請專利範圍第2項所述之處理器，其中，第一邏輯狀態為邏輯真，第二邏輯狀態為邏輯假。

4. 如申請專利範圍第2項所述之處理器，其中，第一邏輯狀態為邏輯假，第二邏輯狀態為邏輯真。

5. 如申請專利範圍第2項所述之處理器，其中，當該指令擷取裝置所擷取之指令為一平行條件執行指令、且該旗標為第一邏輯狀態，該模式切換裝置切換該指令解碼裝置以對該平行條件執行指令之第一N位元指令解碼，以由該指令執行裝置執行該第一N位元指令。

6. 如申請專利範圍第2項所述之處理器，其中，當該指令擷取裝置所擷取之指令為一平行條件執行指令、且該旗標為第二邏輯狀態，該模式切換裝置切換該指令解碼裝置以對該平行條件執行指令之第二N位元指令解碼，以由該指令執行裝置執行該第二N位元指令。

7. 如申請專利範圍第2項所述之處理器，其中，該條件執行指令為M位元指令。

8. 如申請專利範圍第2項所述之處理器，其中，該條件執行指令為N位元指令。

9. 如申請專利範圍第1項所述之處理器，其中，M為32，N為16。

10. 一種於處理器中執行條件式指令之方法，該處理器所執行之指令集包括M位元指令及N位元的指令（M、N為正整數， $M > N$ ），該指令集具有條件執行指令及M位元之平行條件執行指令，該平行條件執行指令具有第一N位元指令及第二N位元指令，該方法主要包含下列步驟：

(A) 擷取至少一指令，以將其解碼並執行之；

(B) 當執行一條條件執行指令時，如該條件執行指令之執行結果為條件成立，設定一旗標之狀態為第一邏輯狀態，如該條件執行指令之執行結果為條件不成立，設定該旗標之狀態為第二邏輯狀態；以及

(C) 當擷取之指令為一平行條件執行指令時，如該旗標為第一邏輯狀態，對該該平行條件執行指令之第一N位元指令解碼並執行之，如該旗標為第二邏輯狀態，則對該平行條件執行指令之第二N位元指令解碼並執行之。

11. 如申請專利範圍第10項所述之方法，其中，第一邏輯狀態為邏輯真，第二邏輯狀態為邏輯假。

12. 如申請專利範圍第10項所述之方法，其中，第一邏輯狀態為邏輯假，第二邏輯狀態為邏輯真。

13. 如申請專利範圍第10項所述之方法，其中，該條件執行指令為M位元指令。

14. 如申請專利範圍第10項所述之方法，其中，該條件執行指令為N位元指令。

15. 如申請專利範圍第10項所述之方法，M為32，N為16。

```

if(R1==0) {R1=R5; R2=R6; R3=R7; R4=R8;}
else      {R1=R9; R2=R10; R3=R11; R4=R12;}
R10 = 0;

```

圖 1

(1)	AL		CMP R1, 0	
(2)	EQ		MOVEQ R1, R5	×
(3)	EQ		MOVEQ R2, R6	×
(4)	EQ		MOVEQ R3, R7	×
(5)	EQ		MOVEQ R4, R8	×
(6)	NE	×	MOVNE R1, R9	
(7)	NE	×	MOVNE R1, R10	
(8)	NE	×	MOVNE R1, R11	
(9)	NE	×	MOVNE R1, R12	
(10)	AL		MOV R10, 0	

↓
[R1]=0

↓
[R1]!=0

圖 2

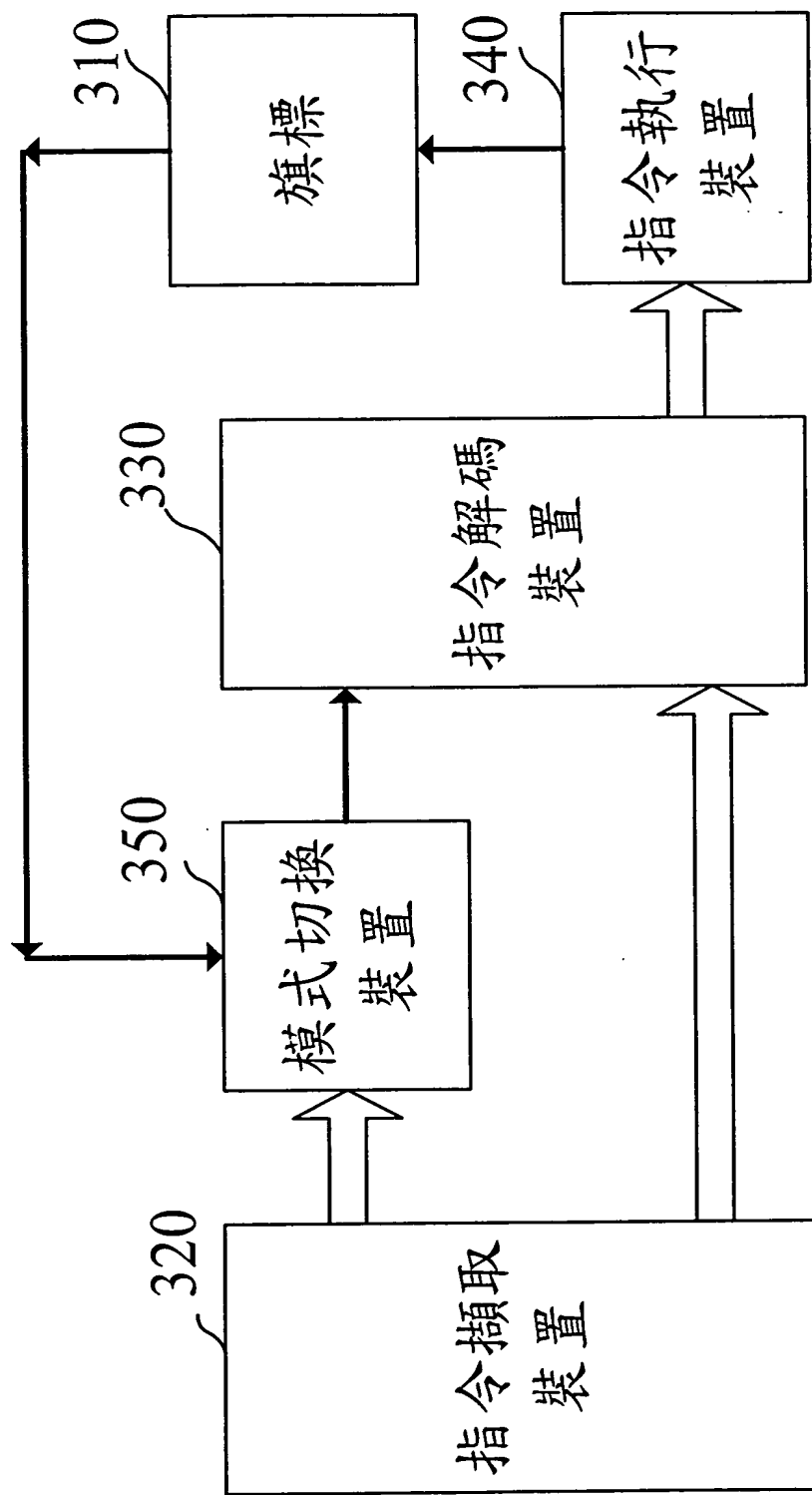


圖 3

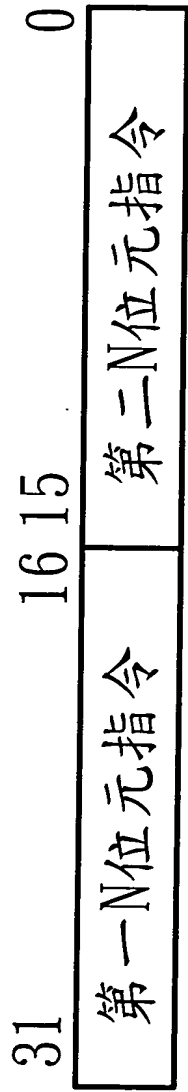


圖 4

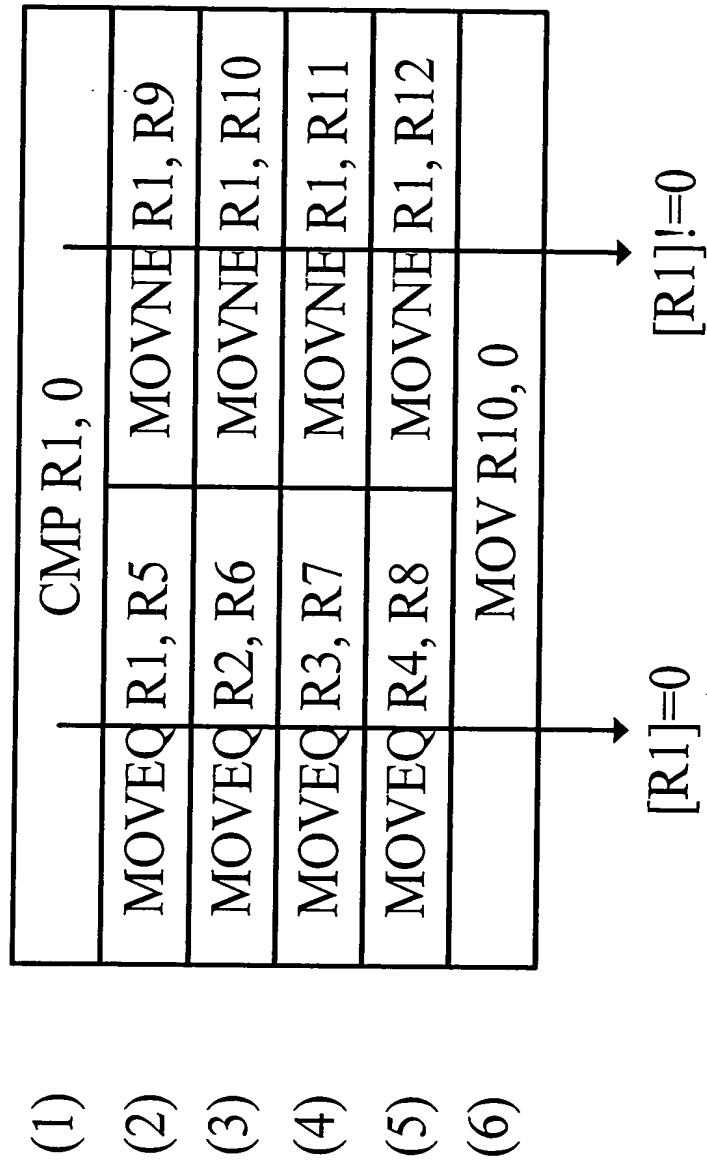


圖 5

(1)	CMP R1, 0	
(2)	other instruction	
(3)	MOVEQ R1, R5	MOVNE R1, R9
(4)	MOVEQ R2, R6	MOVNE R1, R10
(5)	MOVEQ R3, R7	MOVNE R1, R11
(6)	MOVEQ R4, R8	MOVNE R1, R12
(7)	MOV R10, 0	

[R1]=0

[R1]!=0

圖 6

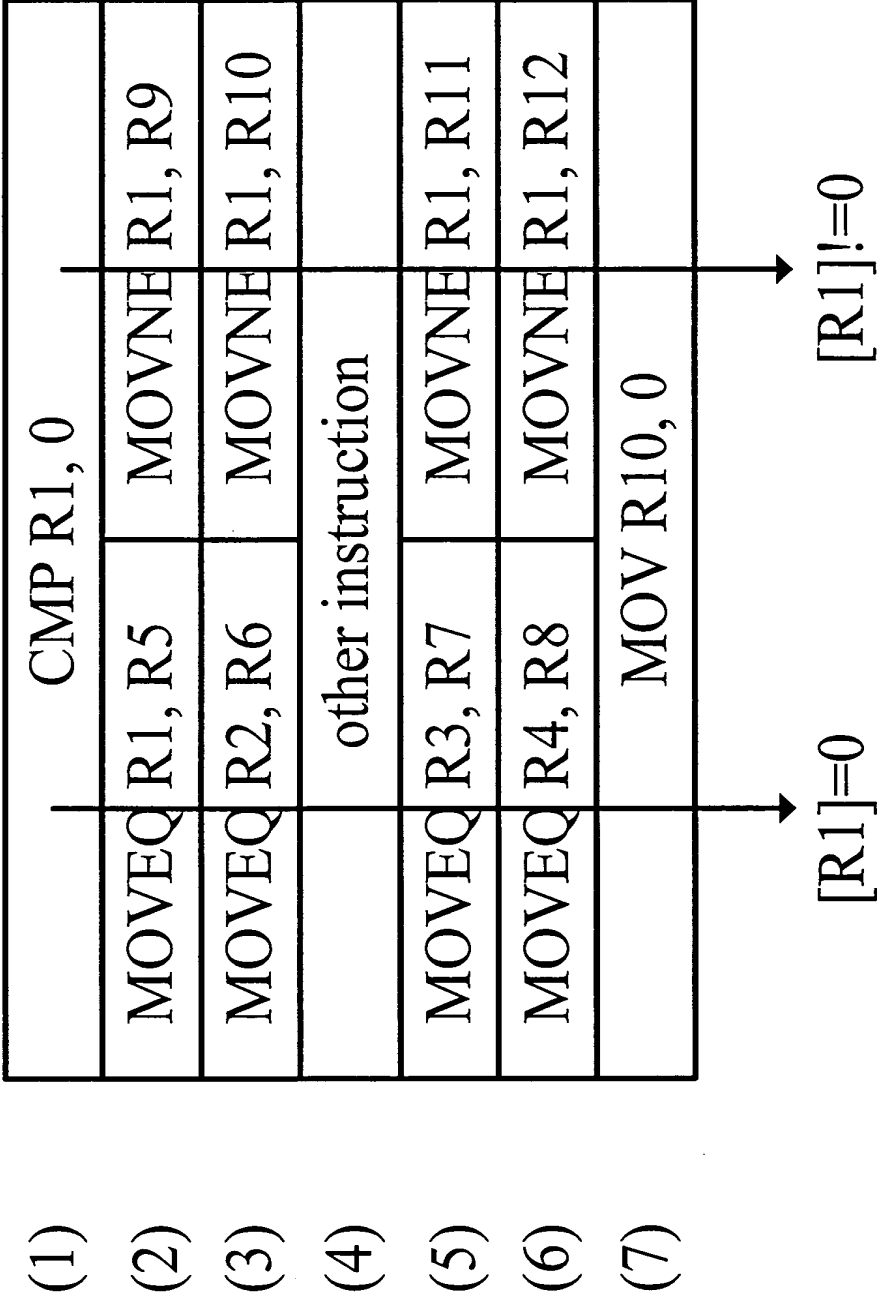


圖 7

other instruction	CMP R1, 0
MOVEQ R1, R5	MOVNE R1, R9
MOVEQ R2, R6	MOVNE R1, R10
MOVEQ R3, R7	MOVNE R1, R11
MOVEQ R4, R8	MOVNE R1, R12
MOV R10, 0	

[R1]=0 [R1]!=0

圖 8